EDB POSTGRES AI

The bug you fixed twice

because no one wrote it down

Lightning talk: PGConf.de 2026

Pinaz Raut
Program Manager I, @EDB
April 22, 2026

Why this belongs at PGConf

Postgres grew because knowledge was shared openly

Postgres didn't grow by people hoarding knowledge. It grew through mailing lists, meetups, and conferences — people sharing not just **code**, but the **context around the code**.

Slack threads

Brilliant fix buried in a private channel.
Gone in a week.

RFC docs

Decision written down.
Reasoning lost with the author.

One person's head

Senior engineer leaves. Six months of
Postgres intuition walks out the door.

The open source community solved this. Your engineering team can too.

The Silo Problem:

Hard problems get solved... quietly

⚠ **A real Postgres example:**

An engineer spends 3 days diagnosing an autovacuum freeze storm on a high-write table. She finds the fix — a targeted storage parameter tweak. Ships it. No write-up. Six months later, a colleague hits the same wall. Three more days lost.

Hard problem solved → **Knowledge stays private** → **Same problem hits again**

This isn't an EDB problem — it's a universal engineering challenge, and especially costly in Postgres teams where deep internals knowledge takes years to develop.

In high-growth teams, brilliant solutions often stay trapped in a single developer's head, RFCs, or a private Slack channel.

The Solution

A format any Postgres team can run

The Format

A platform where engineers share:
- what went wrong
- why it was difficult
- what didn't work
- how it was finally solved

The Scope

Deep dives into Postgres internals, cloud architecture, or transparent breakdowns of

"How we broke (and fixed) X."

The Goal

Excellence isn't an end state; it's a habit.

We are building a culture of continuous sharing and growth.

The Reach

What a talk actually looks like — and why it sticks

Example talk: **"How we broke (and fixed) replication at 3am"**

A cascading failure triggered by a hot standby conflict during a bulk load. The engineer walked through: why wal_receiver_timeout was a red herring, what max_standby_streaming_delay actually does under pressure, and the 2-line config change that prevented it from recurring.

The Library

Every talk is recorded, indexed, and searchable. Not a dead Confluence page.

The Impact

Significantly reducing Time to Productivity for new engineers by providing immediate access to rich historical wisdom.

Live Q&A

Video + Q&A retains the reasoning — the hesitations, the dead ends — that static docs can never capture.

Why it matters

Knowledge sharing across teams

Better engineering communication

Stronger contributions to open source

Collective debugging wisdom

Feedback loop

Encouragement for new speakers

The Open Source Connection

Communities like PostgreSQL grow because people not just share
code but all the **context** along the journey.

These stories grow in **Mailing lists, discussions, meetups and conferences**

A New Addition: Talk Rehearsals

Led by **Floor Drees** — a safe space for engineers selected for external conferences to practice, get peer feedback, and be ready for the global stage.

Excellence isn't an end state.

It's a habit.

1. You don't need a programme. Start with one talk. One honest post-mortem.
2. The Postgres community grows when the stories travel. Help yours travel.
3. The fix you documented today is the 3am panic you prevented next year.

Thank You!